

# PENDETEKSIAN HALANGAN PADA ROBOT CERDAS PEMADAM API MENGGUNAKAN KAMERA DENGAN INTEGRAL PROYEKSI

Setiawardhana<sup>1)</sup>, Nana Ramadijanti<sup>2)</sup>, Rizky Yuniar Hakkun<sup>3)</sup>, Aji Seto Arifianto<sup>4)</sup>

<sup>1,2,3)</sup>Dosen Jurusan Teknik Informatika, <sup>4)</sup>Mahasiswa Jurusan Teknik Informatika

Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember

Kampus ITS Keputih Sukolilo Surabaya 60111 Telp. 031-5947280, Fax 031-5946114

## ABSTRAK

Dalam peraturan Kontes Robot Cerdas Indonesia 2008 khususnya untuk divisi senior beroda, semua peserta diwajibkan untuk mengambil mode furniture. Furniture adalah sebuah halangan berupa tabung dari bahan pipa paralon diisi campuran semen berdiameter 11 cm dengan tinggi 30 cm. Untuk menghindari dinding dan halangan biasanya digunakan sensor jarak, seperti sensor ultrasonik dan untuk mendeteksi api digunakan flame detector, misal sensor UV Tron. Dalam studi ini digunakan kamera sebagai pengganti sensor-sensor analog diatas. Untuk sistem pendeteksian halangan digunakan metode segmentasi warna, integral proyeksi dan diferensial sedangkan untuk mendeteksi api digunakan metode segmentasi warna. Kamera mengirimkan data ke komputer, proses selanjutnya komputer mengirimkan data ke mikrokontroler untuk menggerakkan motor kanan dan motor kiri pada robot. Keberhasilan deteksi halangan dan deteksi api banyak ditentukan oleh tingkat pencahayaan saat dilakukan pengambilan objek

*Kata kunci: integral proyeksi, differensial, segmentasi warna, kamera, dan halangan.*

## 1. PENDAHULUAN

Saat ini robot menjadi salah satu alternatif untuk membantu meringankan tugas manusia, salah satu aplikasinya adalah robot pemadam api, yang nantinya diharapkan dapat dikembangkan untuk menggantikan peran manusia dalam mengatasi masalah kebakaran. Sehubungan dengan itu, maka diselenggarakanlah Kontes Robot Cerdas Indonesia Tahun 2008 (KRCI-2008) oleh DIKTI dengan harapan KRCI-2008 mampu menjadi wahana untuk mendorong kemampuan kreativitas mahasiswa untuk membuat suatu sistem dalam bentuk desain robot cerdas yang mampu menemukan dan memadamkan sumber api, sebagai simulasi penanggulangan dini terhadap bahaya kebakaran.

Dalam Kontes Robot Cerdas Indonesia Tahun 2008 (KRCI-2008) [6] terdapat empat divisi, salah satunya divisi senior beroda. Pada divisi senior beroda tugas utama robot adalah menemukan api lilin dan memadamkannya. Setiap robot peserta dihadapkan dengan beberapa halangan, dan salah satu yang wajib diambil oleh peserta adalah halangan berupa *furniture*. *Furniture* adalah sebuah tabung berdiameter 11 cm tinggi 30 cm, yang diletakkan di ruangan yang ada api lilinnya. Sedangkan metode yang diusulkan pada penelitian akan memberi sumbangan dalam hal navigasi robot untuk menghindari halangan berupa *furniture*.

Prosesnya adalah melalui pengambilan citra secara otomatis dengan kamera, melakukan *thresholding* sehingga diperoleh citra yang hanya mempunyai dua nilai derajat keabuan yaitu hitam dan putih, selain itu metode yang digunakan antara

lain Segmentasi warna [2][3][4][8], integral proyeksi[1][3][4][7], differensial[1] [3][4][7]:

### 1.1. Segmentasi warna

Segmentasi merupakan proses untuk memisahkan objek yang kita ambil dengan latar belakang menggunakan teknik pengelompokan (*clustering*) warna-warna. Segmentasi warna digunakan untuk mendeteksi halangan dan api lilin. Proses segmentasi yang dilakukan dalam studi ini adalah dengan menghitung nilai *euclidean distance* [5].

$$d = \sqrt{(R - \bar{R})^2 + (G - \bar{G})^2 + (B - \bar{B})^2} \dots\dots\dots(1)$$



**Gambar 1.** Segmentasi warna

### 1.2. Integral proyeksi

Integral proyeksi adalah metode yang digunakan untuk mencari batas daerah atau lokasi dari objek.. Metode ini juga bisa disebut dengan integral baris dan kolom dari piksel, karena integral ini menjumlahkan piksel per baris dan piksel per kolom. Integral proyeksi digunakan untuk mencari posisi sudut ruangan terhadap halangan.

$$h_x(k) = \sum_{i=0}^{n_y} I(k, i) \dots\dots\dots(2a)$$

$$h_y(k) = \sum_{i=0}^{n_y} I(i,k) \dots\dots\dots(2b)$$

Dimana:

$h_x(k)$ ,  $h_y(k)$  = masing-masing integral proyeksi terhadap baris x dan terhadap kolom y.

$n_x \times n_y$  = ukuran gambar

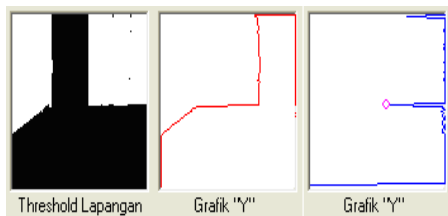
$I(i,k)$  = nilai keabuan pada baris ke- $i$  dan kolom ke- $k$ .

### 1.3. Differensial

Differensial merupakan turunan pertama pada hasil integral proyeksi, sehingga didapatkan tingkat perbedaan antar baris (dari 0 - 255).

$$f^1(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \dots\dots\dots(3)$$

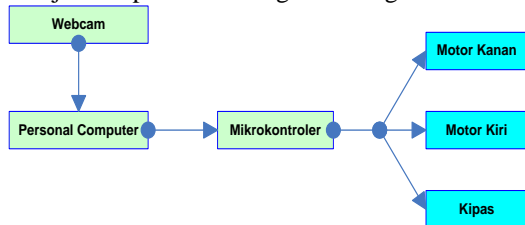
$$y = f(X) + f^d(x).h(x) \dots\dots\dots(4)$$



**Gambar 2.** Threshold, Integral Proyeksi, Differensial

## 2. DISAIN SISTEM

Secara garis besar sistem yang dibangun ditunjukkan pada blok diagram sebagai berikut :



**Gambar 3.** Blok diagram rancangan sistem

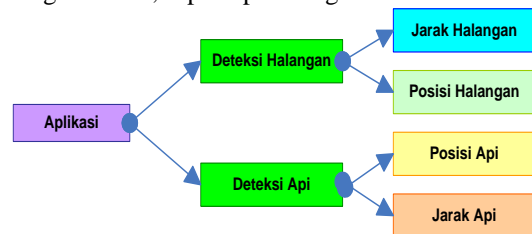
Pembahasan pada makalah ini terbatas pada proses pengambilan input (gambar) oleh kamera, yang pemrosesannya dilakukan di dalam komputer, output yang dihasilkan ditampilkan dalam bentuk simulasi yang ada pada aplikasi yang dibuat.

Algoritma program yang dibuat seperti digambarkan pada diagram alur di bawah ini :



**Gambar 4.** Diagram alur program aplikasi

Dari alur diatas, pembuatan sistem dibagi kedalam 3 bagian besar, seperti pada bagan di bawah ini :

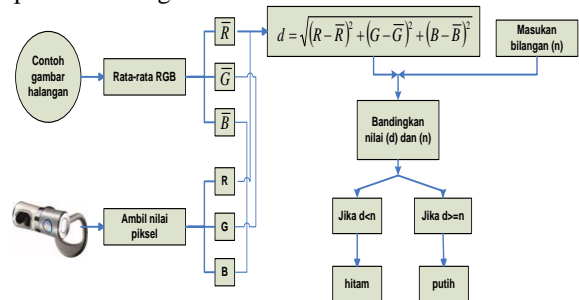


**Gambar 5.** Pembagian sistem kerja aplikasi

### 2.1 Deteksi Jarak Halangan

#### 2.1.1 Segmentasi Warna Halangan

Segmentasi warna api secara umum dijelaskan pada blok diagram berikut ini.



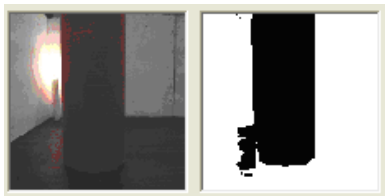
**Gambar 6.** Blok diagram segmentasi Halangan

Terlebih dulu diperlukan contoh warna halangan. Kemudian didapatkan rata-rata fitur warna dari warna halangan.



**Gambar 7.** Contoh dan rata-rata RGB fitur warna halangan

Setelah itu, ditentukan range atau daerah nilai warna halangan. Range warna halangan ditentukan dengan memasukkan nilai batas (n). Kemudian setiap pixel pada gambar dihitung nilai *euclidian distance* (d), seperti yang telah dibahas diatas, dan dibandingkan dengan nilai pembatas (n). Piksel-piksel yang memiliki nilai *euclidian distance* (d) dibawah nilai pembatas (n) akan diberi warna hitam dan sisanya diberi warna putih. Dengan demikian akan terlihat gambar hasil segmentasi warna yang memisahkan objek halangan dengan objek lainnya. halangan akan berwarna hitam dan selainnya akan berwarna putih.



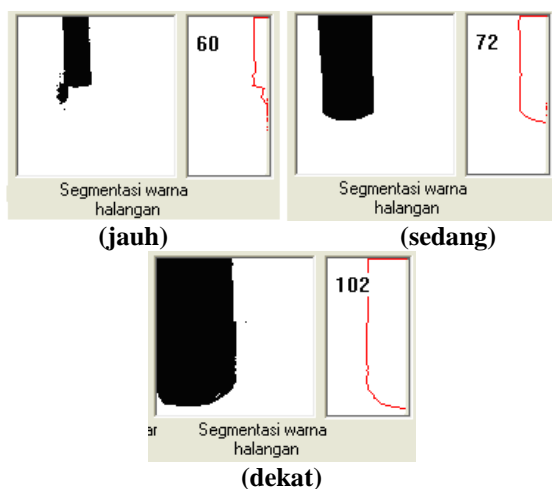
**Gambar 8.** Hasil segmentasi warna halangan.

### 2.3.2 Menentukan jarak robot dengan halangan

Deteksi jarak halangan digunakan agar dapat mengetahui jarak antara robot dengan halangan masih jauh atau sudah cukup, jarak ini digunakan sebagai acuan saat robot akan bergerak menghindari halangan. Secara umum kondisi deteksi jarak halangan dibagi menjadi 3, yaitu jauh, sedang, dan dekat.

Cara menentukan jarak adalah dengan mencari posisi bagian bawah halangan dari proses differensial. Nilai jarak disini dalam satuan piksel

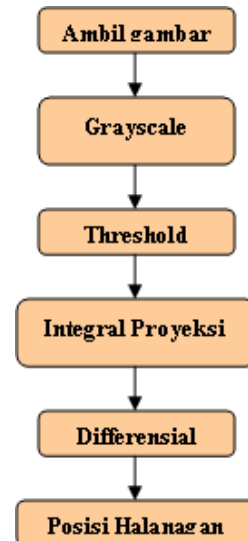
|               |                             |
|---------------|-----------------------------|
| <i>Jauh</i>   | = jarak < 70                |
| <i>Sedang</i> | = jarak >=70 dan jarak <100 |
| <i>Dekat</i>  | = jarak >=100.....(5)       |



**Gambar 9.** deteksi jarak halangan

### 2.3.3 Mencari Posisi Halangan

Deteksi posisi halangan digunakan agar dapat mengetahui posisi halangan terhadap dinding ruangan. Secara umum ada 2 kondisi, yaitu dinding dikiri halangan dan dinding dikanan halangan. Untuk mengetahui posisi halangan secara umum dapat dilihat pada blok diagram berikut ini:

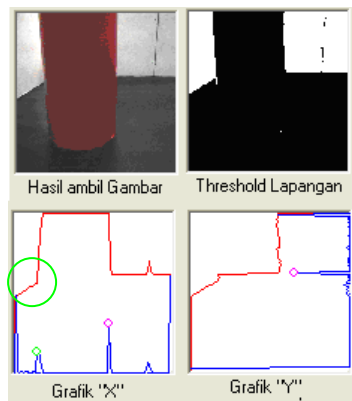


**Gambar 10.** Blog diagram deteksi Posisi Halangan

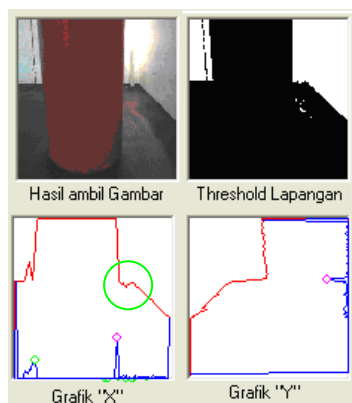
Cara mengetahui posisi halangan terhadap dinding yaitu:

1. Mencari nilai differensial kolom (y) yang maksimum, dan menyimpan iterasi (i) dari nilai maksimum tersebut.
2. Mendapatkan nilai "posisiY" dengan cara, tinggi citra dikurangi dengan nilai iterasi (i) pada poin (1).
3. Mencari posisi halangan, dengan indikasi mendapatkan dua nilai maksimum differensial baris (x).
4. Mencari ramp (grafik miring) pada data integral proyeksi secara horizontal atau terhadap baris (x).
5. Mencari titik sudut ruangan, dengan cara menyeleksi setiap nilai integral proyeksi baris (x) yang mengandung ramp sampai ditemukan nilai integral proyeksi yang sama dengan nilai "posisiY" yang didapat pada poin (2). Nilai inilah yang disebut dengan "posisiX".
6. Jika "posisiX" = 0 (nol) maka dinding berada dikiri halangan. Sebaliknya, posisi halangan maka dinding berada dikanan halangan.

Berikut contoh hasil pendeteksian posisi halangan :



(a)



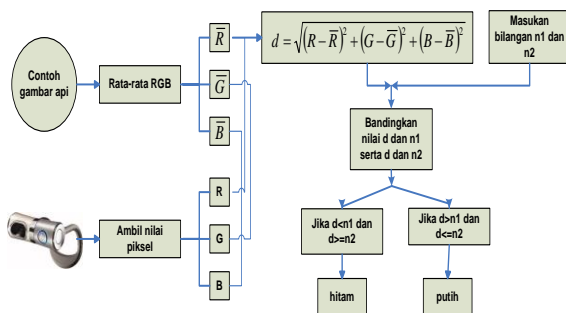
(b)

**Gambar 3.14.** Deteksi posisi halangan: (a) dinding dikiri halangan, (b) dinding dikanan halangan.

## 2.3 Deteksi Api

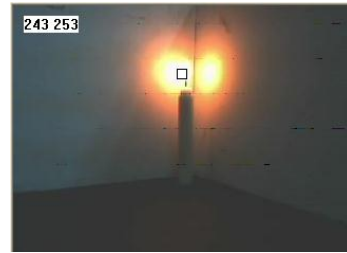
### 2.3.1 Segmentasi Warna Api

Segmentasi warna api secara umum dijelaskan pada blok diagram berikut ini.



**Gambar 9.** Blok diagram segmentasi api

Terlebih dulu diperlukan contoh nilai-nilai fitur warna api. Kemudian didapatkan nilai gray-scale untuk setiap piksel, dari data yang ada didapatkan nilai gray-scale terkecil (min) dan terbesar (maks) untuk nilai pembatas.



**Gambar 10.** Mencari nilai gray-scale fitur warna api

Setelah didapatkan nilai gray-scale minimum dan maksimum, maka nilai tersebut digunakan untuk membatasi nilai piksel gambar.

Kemudian setiap pixel pada gambar dibandingkan dengan range warna api. Jika nilai pixel berada di dalam range warna api maka nilai pixel diubah ke

$$\text{min} < \text{nilai\_api} < \text{maks} \dots \dots \dots (6)$$

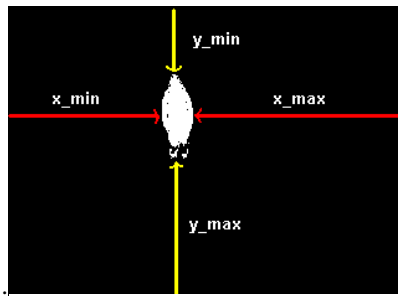
nilai maksimal (warna putih). Jika nilai pixel berada di luar range warna api maka nilai pixel diubah ke nilai minimal (warna hitam). Dengan demikian akan terlihat gambar hasil segmentasi warna api yang memisahkan objek api dengan objek lainnya. Api akan berwarna putih dan selain api akan berwarna hitam



**Gambar 11.** Hasil segmentasi warna api.

### 2.3.2 Mencari Posisi Api

Posisi api digunakan untuk menjaga letak api agar tetap berada di tengah-tengah bidang gambar, hal ini dimaksudkan agar dapat menjaga posisi robot berada tepat di depan api. Penentuan daerah hasil segmentasi warna api dilakukan dengan mencari pixel berwarna putih pertama pada 4 sisi bidang gambar. Sehingga didapatkan nilai koordinat x,y terkecil dan koordinat x,y terbesar.

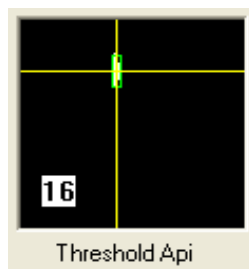


**Gambar 12.** Pencarian posisi api

Setelah posisi api didapatkan, dilakukan perhitungan koordinat pusat dari daerah luasan api dengan rumusan :

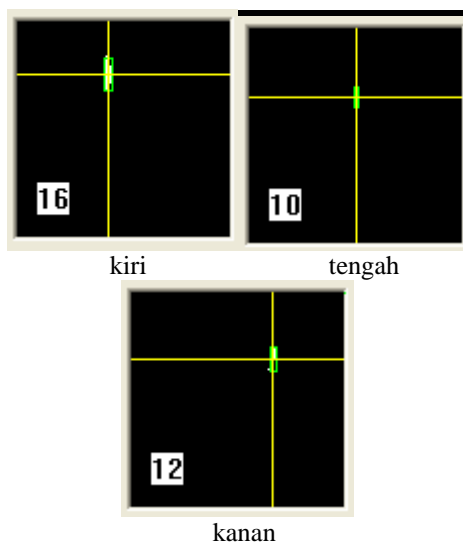
$$\begin{aligned} X \text{ pusat} &= X \text{ terkecil} + (X \text{ terbesar} - X \text{ terkecil}) / 2 \\ Y \text{ pusat} &= Y \text{ terkecil} + (Y \text{ terbesar} - Y \text{ terkecil}) / 2 \dots\dots\dots(7) \end{aligned}$$

Pusat koordinat dari daerah luasan api digunakan untuk menentukan fokus api. Fokus api disini dimaksudkan untuk menempatkan posisi api agar selalu berada di tengah-tengah gambar (persis di depan kamera pada robot).



**Gambar 13.** Penentuan pusat koordinat api

Jika api berada di posisi kiri maka robot akan bergerak ke kiri, dan sebaliknya jika api berada di posisi kanan maka robot akan bergerak ke kanan. Robot akan bergerak lurus jika posisi api berada tepat di tengah gambar.



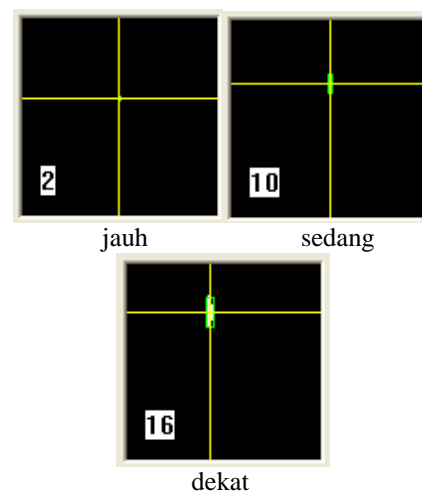
**Gambar 14.** Contoh deteksi posisi api

### 2.3 3 Menentukan Jarak Api dengan Robot

Jarak api terhadap robot digunakan untuk mengetahui posisi robot sudah dekat dengan api atau belum sehingga dapat diambil keputusan kapan saatnya mematikan api.

Jarak api terhadap robot didapatkan dengan cara menghitung daerah luasan api. Sebelumnya harus ditentukan range luasan api untuk kategori **sedang**.

|        |  |
|--------|--|
| Jauh   | : $L_{api} < range\_sedang$                    |
| Sedang | : $L_{api} = range\_sedang$                    |
| Dekat  | : $L_{api} > range\_sedang \dots\dots\dots(8)$ |



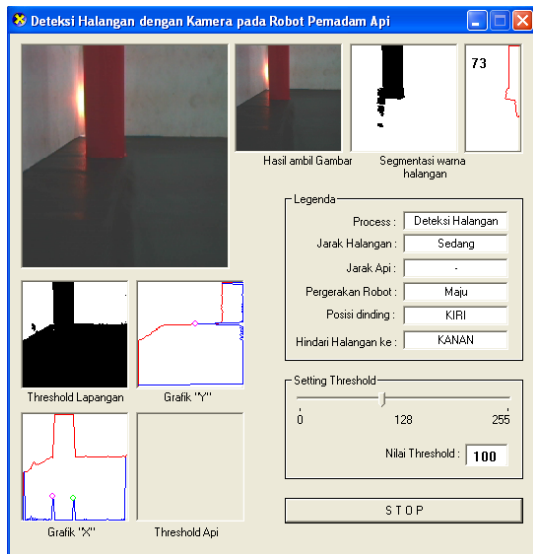
**Gambar 15.** Contoh deteksi jarak api

## 3. PENGUJIAN DAN ANALISA

Pengujian dilakukan sebanyak 60 kali, di hall gedung D4 PENS-ITS. Pengujian dilakukan di dua ruang (R1 dan R3) lapangan KRCI Senior Beroda 2008. Masing-masing pengujian dibagi ke dalam 3 kondisi berdasarkan waktu:

1. Pagi
2. Siang
3. Malam

### 3.1 PENGUJIAN PROGRAM DETEKSI HALANGAN



**Gambar 16.** Tampilan aplikasi saat deteksi jarak halangan.

Dari 1 pengujian **deteksi halangan di Ruang 1** didapatkan data sebagai berikut :

**Tabel 3.1** Prosentase Kebenaran Hasil Uji Coba Halangan Ruang 1

|           | Ruang 1            |                  |                 |
|-----------|--------------------|------------------|-----------------|
| Waktu     | D. posisi halangan | D.jarak halangan | Nilai threshold |
| Pagi      | 100%               | 100%             | 90              |
| Siang     | 80%                | 100%             | 100             |
| Malam     | 100%               | 100%             | 70              |
| Rata-rata | 93.33%             | 100%             |                 |

Dari 1 pengujian **deteksi halangan di Ruang 3** didapatkan data sebagai berikut :

**Tabel 3.2** Prosentase Kebenaran Hasil Uji Coba Halangan Ruang 3

|           | Ruang 1            |                  |                 |
|-----------|--------------------|------------------|-----------------|
| Waktu     | D. posisi halangan | D.jarak halangan | Nilai threshold |
| Pagi      | 100%               | 100%             | 100             |
| Siang     | 100%               | 100%             | 100             |
| Malam     | 100%               | 100%             | 70              |
| Rata-rata | 100%               | 100%             |                 |

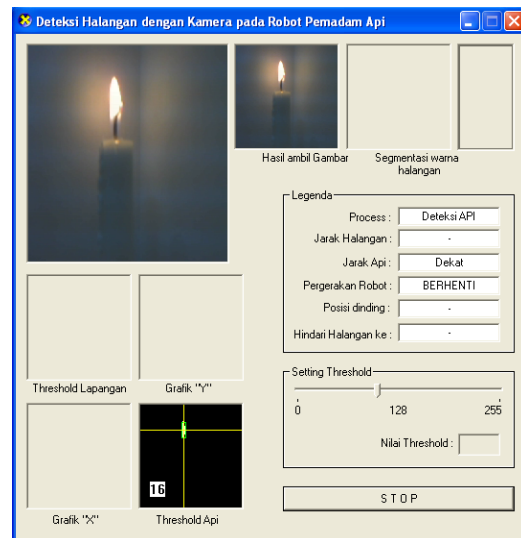
#### Analisa :

- Tingkat pencahayaan yang merata diseluruh bagian objek menjadi faktor utama keberhasilan

pengujian sistem menggunakan kamera ini, seperti halnya pada saat pagi dan malam yang memiliki rata-rata keberhasilan 100%.

- Bayangan dari halangan di dinding yang di akibatkan sumber cahaya dari sisi tertentu berpengaruh terhadap hasil proses threshold. Sehingga bagian yang seharusnya terdeteksi sebagai warna putih bisa saja terdeteksi sebagai warna hitam. Untuk itu dilakukan pembatasan nilai threshold secara dinamis menggunakan tool slider, yang bisa dirubah sesuai kondisi pencahayaan.
- Saat siang hari hasil percobaan cenderung jelek, karena intensitas cahaya yang mengenai objek tidak merata, Bagian kakan lapangan terkena cahaya cukup terang, sehingga terbentuk bayangan halangan didinding dan tidak dapat diproses menjadi citra biner secara sempurna, akibatnya sistem tidak berhasil mendapatkan posisi ramp dengan benar.

### 3.2 PENGUJIAN PROGRAM DETEKSI API



**Gambar 17.** Tampilan aplikasi saat deteksi api

Dari 1 pengujian **deteksi Api di Ruang 1** didapatkan data sebagai berikut :

**Tabel 3.3** Prosentase Kebenaran Hasil Uji Coba Api Ruang 1

|           | Ruang 1       |             |                 |
|-----------|---------------|-------------|-----------------|
| Waktu     | D. posisi api | D.jarak api | Nilai threshold |
| Pagi      | 100%          | 30%         | 80              |
| Siang     | 100%          | 60%         | 80              |
| Malam     | 100%          | 30%         | 80              |
| Rata-rata | 100%          | 40%         |                 |



Dari 1 pengujian **deteksi Api di Ruang 3** didapatkan data sebagai berikut :

**Tabel 3.4** Prosentase Kebenaran Hasil Uji Coba Api Ruang 3

|           | Ruang 1       |             |                 |
|-----------|---------------|-------------|-----------------|
| Waktu     | D. posisi api | D.jarak api | Nilai threshold |
| Pagi      | 100%          | 50%         | 80              |
| Siang     | 100%          | 50%         | 80              |
| Malam     | 100%          | 30%         | 80              |
| Rata-rata | 100%          | 43.33%      |                 |

#### Analisa :

- Untuk deteksi posisi api secara umum tidak berpengaruh pada tinggi atau rendahnya tingkat pencahayaan, karena api sendiri merupakan objek yang memancarkan cahaya. Dan besar kecilnya api yang terdeteksi tidak berpengaruh, karena deteksi posisi api berdasarkan titik tengah api dibandingkan dengan titik tengah frame.
- Hal yang paling berpengaruh pada proses deteksi api adalah resolusi dari kamera yang dipakai. Pada studi ini dipakai kamera dengan resolusi video 960 x 720 piksel, namun memiliki kelemahan saat deteksi api di malam hari, bayangan api yang terpancar ke dinding tertangkap kamera sehingga menghasilkan bentuk api yang tidak sempurna dan relatif besar, hal ini mengakibatkan robot sudah mendeteksi lilin ada pada jarak dekat, padahal sebenarnya jarak robot terhadap lilin masih jauh atau sedang.

### 3.3 Analisa terhadap Proses Segmentasi Warna Halangan

Ada dua cara yang penulis lakukan untuk proses segmentasi warna halangan :

1. Dengan menghitung nilai euclidean distance antara nilai rata-rata RGB halangan yang di dapat dari data contoh (tidak realtime) dengan nilai RGB setiap piksel pada citra dalam kondisi realtime.



**Gambar 18.** Hasil segmentasi warna dengan euclidean distance

2. Dengan pembatasan biasa, artinya berdasarkan asumsi bahwa nilai warna

merah (red) pada citra halangan selalu lebih besar dari pada nilai hijau (green) dan biru (blue), serta pembatasan nilai warna merah (red) minimal dari halangan yang didapat dari data contoh.

Gambar 4.5 dan 4.6 menunjukkan contoh hasil segmentasi warna dengan pembatasan biasa

Jika fitur merah lebih besar dari 80 dan merah lebih besar dari hijau dan biru.



**Gambar 19.** Hasil segmentasi warna dengan pembatasan nilai merah lebih besar dari 80.



**Gambar 20.** Hasil segmentasi warna dengan pembatasan nilai merah lebih besar dari 120.

#### Analisa

Dari dua cara diatas cara no (1) lebih efektif karena hasilnya lebih akurat. Dengan catatan pengambilan data contoh diambil didaerah yang paling bagus warnanya sehingga nilai rata-rata fitur warnanya bisa merepresentasikan warna objek tersebut. Yang dimaksud dengan daerah yang paling bagus biasanya berada dibagian tengah objek dan pengambilan gambar contoh tidak menggunakan lampu kamera yang terang (blitz).

Cara kedua relatif lemah, karena dengan cara tersebut, proses segmentasi masih bisa menangkap noise-noise disekitar objek. Disekitar objek terdapat daerah-daerah yang merekam pantulan warna dari objek, contohnya dinding, sehingga guratan-guratan warna merah halangan yang ada didinding tertangkap oleh proses segmentasi warna

### 4. KESIMPULAN

Setelah dilakukan pengujian dan analisisnya, dapat diambil beberapa kesimpulan antara lain :

1. Tingkat keberhasilan deteksi jarak halangan mencapai **100%**, dipengaruhi oleh hasil segmentasi warna halangan. Proses segmentasi warna dengan menghitung nilai euclidean

distance sangat baik, namun sangat tergantung dengan nilai rata-rata fitur yang didapat dari data contoh.

2. Keberhasilan deteksi posisi halangan terhadap dinding mencapai **93.33%**, faktor paling berpengaruh adalah intensitas cahaya yang mengenai objek, intensitas cahaya semakin merata hasil semakin bagus. Namun kelemahan ini dapat diatasi dengan pemberian batas nilai threshold yang dinamis, sehingga ketika kondisi lingkungan percobaan berubah tinggal mengatur nilai pembatas threshold.
3. Tingkat keberhasilan deteksi posisi api hingga mencapai **100%** sangat ditentukan oleh rata-rata sampel gambar api dan pemberian nilai toleransi pada warna api saat proses segmentasi api.
4. Jenis kamera yang digunakan dan intensitas cahaya pada lingkungan berpengaruh terhadap buruknya hasil deteksi jarak api yang hanya berkisar lebih kurang **40%**. Karena saat cahaya tidak merata kamera dengan resolusi tinggi akan menangkap pendaran api lilin didinding.

## 5. DAFTAR PUSTAKA

- [1] Basuki, Achmad., Nana Ramadiajnti, Tri Harsono. *Deteksi Rambu-Rambu Batas Kecepatan menggunakan filter RGB dan Integral Proyeksi*. Surabaya: IES, EEPIS.2007.
- [2] Batavia, Parag H., Sanjiv Singh. *Obstacle Detection Using Adaptive Color Segmentation and Color Stereo Homography*. Pittsburgh: Carnegie Mellon University, Robotics Institute.2001.
- [3] Besari, Adnan R.A., Setiawardhana, Riyanto Sigit, Dadet Pramadihanto. *Robot Cerdas Pemadam Api menggunakan Kamera*. IES, EEPIS 2006.
- [4] Gonzalez, Rafael C., Woods, Richard C. *Digital Image Processing*. Prentice Hall. 2nd edition. 2002.
- [5] Hamdhani R.M. *Robot Pemindah Objek Dengan Kemiripan Bentuk Sub Judul : Deteksi Fitur Bentuk*. Surabaya : Politeknik Elektronika Negeri Surabaya – ITS.2007.
- [6] Panitia KRCI 2008 DP2M Dikti. *Kontes Robot Cerdas Indonesia 2008 Robot Cerdas Pemadam Api*.Jakarta: DIKTI.2007.
- [7] Setiawardhana, Riyanto S., Dadet P. *Robot Cerdas Pemadam Api menggunakan Kamera dengan Logika Fuzzy*. ICICI , ITB 2007.
- [8] Sigit, Riyanto. *Modul praktikum Image Processing*. Surabaya : Politeknik Elektronika Negeri Surabaya – ITS. 2005.